SISTEM OTENTIKASI UNTUK SQUID BERBASIS WEB

Febriliyan Samopa -- Royyana M.I. -- Liga Awami
Program Studi Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.
Email : iyan@its-sby.edu
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.
Email: roy@inf.its.ac.id,liga@it-7.com

ABSTRAK

Dalam sebuah jaringan komputer terdapat bermacam-macam tipe user dengan berbagi tingkatan yang berbeda yang juga dibutuhkan perlakuan yang berbeda pada tiap user yang disesuaikan dengan kebutuhan dalam menggunakan akses web, baik berupa http, ftp, gopher, dan lain-lain. Dimana masing-masing user memiliki skala prioritas dalam penggunaan bandwidth, jumlah koneksi maksimum, waktu koneksi, ukuran file maksimum, situs yang tidak boleh diakses dan lain-lain. Tujuan dari penelitian ini adalah membuat sebuah perangkat lunak yang dapat melakukan otentikasi user berdasarkan data konfigurasi yang disimpan dalam basisdata. Selain itu, perangkat lunak yang dibuat dapat memproses request dari client berdasarkan data konfigurasi dengan lebih cepat. Permasalahan yang mucul adalah bagaimana merancang dan membuat suatu perangkat lunak yang dapat melakukan otentikasi user berdasarkan data konfigurasi yang diambil dari basisdata, serta dapat memproses request dari client dengan lebih cepat berdasarkan hak akses yang dimilikinya.

Dalam penelitian ini didesain dan diimplementasikan suatu sistem otentikasi user dengan mengambil data user yang tersimpan dalam basisdata MySQL. Disamping itu, dilakukan rekayasa pada beberapa rutin proses yang terdapat dalam squid proxy, supaya proses-proses dapat melakukan pengambilan data konfigurasi yang dialihkan dan disimpan dalam basisdata MySQL. Data konfigurasi ini didasarkan pada pembagian hak akses yang dimiliki oleh masing-masing grup user. Antarmuka berbasis web digunakan sebagai salah satu layanan bagi admin untuk mempermudah pengelolaan dan pengolahan data konfigurasi yang dibuat.

Berdasarkan uji coba yang telah dilakukan, terbukti sistem yang dibuat dapat bekerja dengan baik dan tidak melenceng dari fungsi asli sebelum dilakukan perubahan. Bahkan pada penanganan request client yang berukuran besar, kinerja squid mengalami peningkatan dalam hal kecepatan proses yang dibutuhkan. Sebagai contoh, request client pada file berukuran 602 KB dengan tipe file html (supaya dapat dieksekusi dan ditampilkan pada browser). Pada squid asli, waktu rata-rata (dari lima kali percobaan) yang dibutuhkan untuk menyelesaikan pengiriman data yang diminta adalah 85,921 detik. Sedangkan pada squid hasil rekayasa, waktu rata-rata (dari lima kali percobaan) yang dibutuhkan adalah 76,572 detik. Dengan demikian dapat diambil kesimpulan, dengan pembuatan rekayasa rutin proses yang tepat dapat meningkatkan kinerja squid.

Kata Kunci : *squid, proxy, otentikasi eksternal mysql, linux, web, php, rekayasa proses, patch, akses user*

1. PENDAHULUAN

Dalam sebuah jaringan komputer terdapat bermacam-macam tipe *user* dengan berbagi tingkatan yang berbeda. Disini juga dibutuhkan perlakuan yang berbeda pada tiap *user* yang disesuaikan dengan kebutuhan dalam menggunakan akses *web*, baik berupa *http*, *ftp*, *gopher*, dan lain-lain. Dimana masing-masing *user* memiliki skala prioritas dalam penggunaan *bandwidth*, jumlah koneksi maksimum, waktu koneksi, ukuran file maksimum, situs yang tidak boleh diakses dan lain-lain.

Tujuan penelitian ini adalah membuat sebuah perangkat lunak yang dapat melakukan otentikasi *user* berdasarkan data konfigurasi yang disimpan dalam basisdata. Dalam hal ini, data konfigurasi dapat tersimpan lebih aman dari akses *user* yang tidak berhak, meskipun hanya untuk membaca data tersebut. Selain itu, sistem yang dibuat dapat memproses *request* dari *client* berdasarkan data konfigurasi dengan lebih cepat.

Permasalahan yang diangkat dalam penelitian ini adalah bagaimana merancang dan membuat suatu sistem yang dapat melakukan otentikasi *user* berdasarkan data konfigurasi yang diambil dari basisdata, serta dapat memproses *request* dari *client* dengan lebih cepat berdasarkan hak akses yang dimilikinya.

Sistem yang akan dirancang dalam penelitian ini memiliki batasan-batasan sebagai berikut:

1). Sistem yang dihasilkan dari penelitian ini adalah sistem yang berbasis Linux;
2). Sistem yang dibangun dapat mengidentifikasi tingkatan *user* berdasarkan grup *user* yang menggunakan akses jaringan;
3). Sistem yang dibangun dapat melakukan penambahan, perubahan dan penghapusan terhadap data konfigurasi penggunaan jaringan yang digunakan *user*;
4). Sistem yang dibangun dapat mengenkripsi data yang bersifat rahasia seperti password.

5). Sistem yang dibangun dapat mengkonfigurasi aturan yang dipakai untuk tiap grup *user*, yaitu jumlah koneksi maksimum, ukuran file maksimum yang bisa di-*download*, waktu koneksi yang diijinkan, alamat *url* yang tidak boleh diakses dan besar *bandwidth* yang diijinkan.
6). Tools yang digunakan untuk pembuatan sistem ini adalah:
   - Sistem Operasi Linux Mandrake 9.1.
   - GCC GNU Linux.
   - Server Web Apache.
   - Server MySQL
   - Server Squid.

Manfaat pertama yang didapat dengan dibuatnya sistem ini adalah setiap *user* memiliki konfigurasi masing-masing berdasarkan data konfigurasi yang dimiliki ketika terhubung ke *proxy server* squid, dimana data konfigurasi yang disimpan dalam basisdata dapat terjaga dengan lebih aman.

Yang kedua, proses pelayanan *request* dari *client* dapat dilakukan dengan lebih cepat. Dengan peningkatan kecepatan pemrosesan *request client*, maka kinerja proxy *server* squid juga semakin meningkat. Hal ini tentunya dapat lebih mengurangi kepadatan lalu lintas jaringan komputer, sehingga dapat dikurangi *collison* yang terjadi pada komunikasi data yang melewati jaringan. Dengan demikian kualitas lalu lintas jaringan dapat ditingkatkan

2. SQUID

Squid adalah sebuah *cache proxy server* dengan kinerja tinggi untuk web client, mendukung objek data FTP, gopher, dan HTTP. Tidak seperti *caching software* tradisional, squid menangani banyak request dalam sebuah proses *single, non-blocking, I/O-driven process*.

Squid menyimpan meta data dan khususnya data yang sering digunakan dalam RAM, *cache DNS lookups*, dukungan *non-blocking DNS lookups*, dan menerapkan *negative caching* dari *object request* yang gagal.

Squid mendukung SSL, *extensive access controls*, dan *full request logging*. Dengan menggunakan Internet *Cache Protocol* (ICP), squid *cache* dapat disusun kembali dalam sebuah hirarkhi untuk menambah penyimpanan bandwidth.

Squid terdiri dari program server utama squid, sebuah program *Domain Name System lookup dnsserver*, beberapa program pilihan untuk menulis kembali *request* dan melakukan otentikasi, dan beberapa *management* dan *client tool*. Ketika squid dijalankan, dihasilkan sebuah nomor yang dapat dikonfigurasi dari proses dnsserver, yang masing-masing dapat melakukan *single, blocking Domain Name System (DNS) lookup*. Hal ini dapat mengurangi lama waktu *cache* ketika menunggu *DNS lookups*.

Internet object caching merupakan cara untuk menyimpan internet object yang diminta (seperti, data yang tersedia melalui protokol HTTP, FTP, dan gopher) pada sebuah sistem yang lebih dekat pada situs yang diminta dibandingkan sumber. Web browser dapat menggunakan *cache* squid lokal sebagai HTTP proxy server, menghemat waktu akses sesuai dengan konsumsi bandwidth. Client melakukan permintaan sebuah internet object dari *caching proxy*; jika object tersebut belum ada di-*cache*, proxy server mengambil object tersebut (baik dari host yang dimaksud pada URL atau dari *parent* atau *sibling cache*) dan mengirimnya ke *client*.

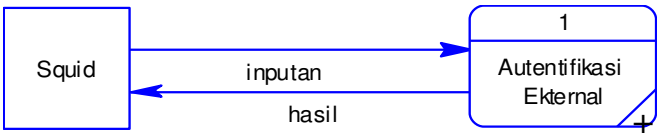
Protokol yang digunakan *cache* squid adalah ICP. ICP utamanya digunakan dengan hirarki *cache* untuk menempatkan object yang ditentukan dalam sibling *cache*. Jika *cache* squid tidak memiliki dokumen yang diminta, dia mengirimkan query ICP pada sibling *cache*, dan sibling merespon dengan jawaban ICP ditandai dengan "HIT" atau "MISS". Kemudian *cache* menggunakan jawaban tersebut untuk memilih dari *cache* mana untuk mengatasi "MISS" tersebut.

ICP juga mendukung banyak transmisi stream object multiplexed melalui koneksi TCP tunggal. ICP sekarang ini diterapkan pada UDP. Versi squid saat ini juga mendukung ICP via multicast.

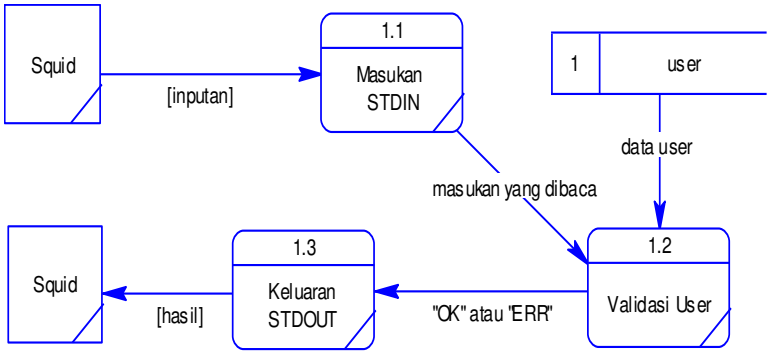
3. OTENTIKASI DASAR SQUID

User akan diotentikasi jika squid dikonfigurasi untuk menggunakan *acl proxy_auth*. Browser mengirim otentikasi user pada *request authorization header*. Jika squid menerima sebuah *request* dan daftar aturan *http_access* menerima sebuah *acl proxy_auth*, squid memeriksa *authorization header*. Jika *header* ada, squid men-*decode*-nya dan mengekstrak username dan password. Jika header tidak ada, squid akan mengembalikan sebuah HTTP reply dengan status 407 (*Proxy Authenticate Required*). Browser menerima *reply 407* dan meminta user untuk memasukkan username dan password. Username dan password di-*encode*, dan dikirim pada *authorization header* untuk *request* berikutnya pada proxy. Untuk lebih jelasnya dapat dilihat pada DAD Tingkat 0 dan 1 yang terdapat pada gambar 1 dan 2.

Modul otentikasi menerima inputan berupa string pada STDIN yang berisi data *username* dan *password*, dalam format *"<username> <password>"* . Dari sini data username dan password diolah lebih lanjut pada proses validasi. Dari proses validasi ini, nantinya akan dikembalikan string, dengan nilai *"OK"* untuk validasi yang berhasil atau *"ERR"* untuk validasi yang gagal, melalui STDOUT.



Gambar 1. DAD Tingkat 0 Cara Kerja Otentikasi



Gambar 2. DAD Tingkat 1 Cara Kerja Otentikasi

4. PROSES OTENTIKASI EKSTERNAL DENGAN MYSQL

Proses ini diperlukan sebagai proses validasi user yang akan menggunakan layanan squid sebagai web *cache* proxy supaya dapat terhubung ke jaringan internet. Dengan adanya proses otentikasi ini akan membatasi ruang gerak user yang tidak berhak dalam menggunakan layanan squid. Pada proses otentikasi ini data user yang boleh menggunakan layanan squid, disimpan dalam MySQL

Langkah pertama adalah membaca setting untuk koneksi ke MySQL yang tersimpan dalam sebuah file teks. Setting koneksi yang dimaksud adalah lokasi server MySQL, nama basisdata, username dan password untuk koneksi MySQL. Setelah data konfigurasi koneksi didapatkan, dilanjutkan dengan koneksi ke server MySQL.

Jika koneksi gagal akan menghasilkan nilai kembalian berupa string "ERR" dan langsung dikembalikan ke server squid melalui STDOUT. Namun sebaliknya, jika koneksi ke MySQL berhasil, dilanjutkan dengan proses query validasi username dan password yang diterima setelah di-parsing, dengan data yang tersimpan dalam MySQL. Jika terdapat data username dan password yang dimaksud dalam MySQL, maka nilai kembalian berupa string *"OK"*. Jika data yang dimaksud tidak ada atau tidak cocok, maka nilai kembalian berupa string *"ERR"*. Nilai kembalian yang berupa string *"OK"* atau *"ERR"* ini dikembalikan ke squid melalui STDOUT.

5. REKAYASA PROSES PEMBATASAN UKURAN FILE, BANDWIDTH, JUMLAH KONEKSI, WAKTU KONEKSI DAN URL YANG BISA DIAKSES

Pembatasan ini dilakukan untuk mengurangi kepadatan jalur koneksi antara proxy squid dengan internet yang terlalu lama, akibat dari request client yang terlalu besar. Secara standar, squid sudah dapat menangani hal ini namun data yang diambil masih dari file konfigurasi squid.conf. Sehingga jika terjadi perubahan konfigurasi maka diperlukan proses *reconfigure* atau *restart* pada server squid. Hal ini memerlukan sedikit waktu tambahan yang dibutuhkan oleh squid untuk membaca ulang data konfigurasi setelah dilakukan perubahan pada data tersebut.

Untuk itu, setting konfigurasi dirubah dan dialihkan pada basisdata MySQL. Ukuran file yang dapat di-*request* disimpan dalam MySQL, berdasarkan konfigurasi yang dilakukan pada masing-masing grup user.

Salah satu keuntungan yang bisa didapat dari rekayasa rutin proses di atas adalah pengambilan data yang selalu terbaru, sehingga selalu mendapatkan nilai data yang terbaru sesaat setelah terjadi perubahan data ukuran file maksimum yang diijinkan, tanpa perlu melakukan reconfigure atau restart pada server squid.

6. ANTARMUKA BERBASIS WEB

Untuk lebih memudahkan pengelolaan sistem dan data pada MySQL diperlukan sebuah sarana perantara, yaitu sebuah antarmuka. Perancangan antarmuka yang sangat lebih cocok digunakan disini adalah sebuah antarmuka berbasis *web*. Antarmuka berbasis *web* dipilih karena kemudahannya dalam

penggunaan, serta dapat diakses dari mana pun, yang mempunyai hak akses terhadap data yang bersangkutan. Pada antarmuka berbasis web ini terdapat beberapa halaman yaitu:

- Halaman Login, digunakan sebagai gerbang untuk dapat mencegah *user* yang tidak berhak terhadap manipulasi sistem dan data.
- Halaman Sistem, digunakan untuk mengontrol beberapa sistem yang berhubungan dengan squid.
- Halaman Data Grup User, digunakan untuk mengolah data grup *user*.
- Halaman Data User, digunakan untuk mengolah data *user*.
- Halaman Data Waktu Koneksi, digunakan untuk mengolah data waktu koneksi.
- Halaman Data Kategori Alamat URL, digunakan untuk mengolah data kategori alamat *url*.
- Halaman Data Alamat URL Regex, digunakan untuk mengolah data alamat *url regex*.
- Halaman Detil Konfigurasi Grup User, digunakan untuk mengolah detil konfigurasi grup *user*.

7. HASIL UJI COBA

Uji coba dilakukan hanya pada lingkungan *Local Areal Network*. Hal ini dimaksudkan untuk mendapatkan kondisi ideal dari setiap skenario uji coba yang dilakukan. Masing-masing proses uji coba dilakukan dengan menggunakan dua macam squid, yaitu squid asli dan squid yang telah dilakukan rekayasa terhadap beberapa rutin proses yang ada dalam squid, dimana seluruh data diambil dari basisdata MySQL.

Sedangkan setting jaringan yang digunakan pada server squid adalah sebagai berikut:

- Alamat IP 10.126.11.105
- Gateway 10.126.11.1
- Primary DNS 202.155.84.178
- Secondary DNS 202.155.84.179
- Parent Proxy 202.155.84.180:8080
- Setting *cache_peer* pada file *squid.conf* adalah *cache_peer 202.155.84.180 parent 8080 0 login=umum:umum proxy-only*

Berikut ini adalah data konfigurasi yang digunakan dalam keseluruhan uji coba:

Uji coba otentikasi user dilakukan dengan menggunakan modul otentikasi eksternal yang mengambil data *user* dari MySQL (Tabel 1), dilakukan uji coba otentikasi *user* yang berhak menggunakan layanan squid.

Uji coba keamanan data konfigurasi dilakukan untuk mendapatkan informasi tentang keamanan data konfigurasi dari *user* yang tidak berhak, meskipun hanya sekedar melihat sekalipun. Yang pertama, uji coba dilakukan dengan melakukan akses pada database *squid_mysql* melalui *shell command*, dengan menggunakan berbagai tipe user dan privilege yang ada. Akses yang dilakukan adalah melakukan perintah *select, insert, update, delete, drop,* dan *alter*. Koneksi ke server MySQL dilakukan melalui shell console. Setelah itu barulah dilakukan uji coba pengaksesan data yang diinginkan.

Tabel 1. Pengelompokan User

Username	Password	Nama Grup	Status
atlanta	atlantapass	MhsS1	enable
mbix	mbixpass	MhsS2	enable
siera	sierapass	MhsS1Ext	enable
suway	suwaypass	MhsS1	disable
pioneer	pioneerpass	MhsS2	disable
sprint	sprintpass	MhsS3	disable

Tabel 2. Keterangan Alamat Url Lokal

Url	Alamat	Ukuran
1	http://10.126.11.246/book-full.html	426 KB
2	http://10.126.11.246/manual_toc.html	137 KB
3	http://10.126.11.246/setuplog.txt	602 KB
4	ftp://squid:squidsquid@10.126.11.105	-
5	http://10.126.11.246/phpinfo.php	47 KB

Tabel 3. Hasil Uji coba Otentikasi User dengan MySQL

Masukan Username	Masukan Password	Hasil	Ket.
atlanta	atlantapass	berhasil	-
atlanta	atlanta	gagal	password salah

atlanta	atlantaPass	gagal	password case sensitive
atlanta	<NULL>	gagal	password salah <NULL>
suway	suwaypass	gagal	user di-disable
sprint	sprintpass	gagal	grup user di-disble

Tabel 4. Hasil Uji Coba Keamanan Melalui Shell Command

User	sel	ins	upd	del	drop	alt
root	v	v	v	v	v	v
squidviewer	v	x	x	x	x	x
squidadmin	v	v	v	v	x	x
user lain	x	x	x	x	x	x

Uji coba keamanan data yang kedua melalui antarmuka web, dilakukan dengan terlebih dahulu memasukkan *username* dan *password* pada halaman login yang sudah disediakan. Satu-satunya gerbang pengaksesan data melalui antarmuka web hanya berupa masukan *username* dan *password* yang benar. Selain itu, penggunaan *session* dapat menunjang keamanan yang terjadi selama proses transaksi berlangsung.

Uji coba kecepatan proses *request client* dilakukan untuk mengetahui seberapa lama waktu yang dibutuhkan server squid ketika melayani *request*. Waktu dihitung mulai dari *client* mengirim *request* sampai dengan *request* tersebut dilayani, baik *request* yang diijinkan maupun *request* yang ditolak.

Proses uji coba dengan melakukan *request* alamat *url* lokal (Tabel 2) yang dikirim oleh *client* ke server squid. Alamat *url* yang dimaksud harus dapat dieksekusi dan ditampilkan pada halaman browser, seperti tipe html, php, teks dan lain-lain. Lama waktu yang dibutuhkan dalam uji coba ini, dihitung mulai dari *client* mengirim *request* sampai dengan hasil reply dari squid yang dikirim ke *client* telah selesai. Uji coba dilakukan dengan konfigurasi *client* sebagai berikut:

- Alamat IP : 10.126.11.107
- Restore Bandwidth : 8000 bytes
- Maks Bandwidth : 8000 bytes
- Maks Koneksi : 6
- Ukuran File Maks : 1000000 bytes

Url	Waktu Rata-rata	
	Squid Asli	Squid Rekayasa
1	63,097 detik	60,841 detik
2	19,726 detik	21,725 detik
3	85,921 detik	76,572 detik
4	12,186 detik	12,172 detik
5	6,678 detik	6,998 detik

8. SIMPULAN DAN SARAN

Berdasarkan pada perancangan dan pembuatan sistem terhadap permasalahan yang diangkat, maka dapat diambil kesimpulan sebagai berikut:

- Program otentikasi eksternal, dengan menggunakan MySQL sebagai tempat penyimpanan data *user*, dapat dijadikan modul otentikasi pada server squid. Selain itu, MySQL juga mendukung enkripsi hash MD5 yang dapat digunakan untuk mengenkripsi data *password* user.
- Rekayasa rutin proses yang dilakukan pada server squid dapat meningkatkan kecepatan pemrosesan request *client* secara keseluruhan, pada ukuran request yang besar.
- Penggunaan antarmuka berbasis web, sebagai media perantara untuk mengatur data konfigurasi yang digunakan, dapat menunjang efektivitas pengaturan data konfigurasi. Hal ini disebabkan antarmuka berbasis web dapat dieksekusi melalui browser dengan konfigurasi standar.

Berikut ini adalah saran untuk kemungkinan pengembangan lebih lanjut dari sistem ini:

- Efektivitas rekayasa rutin proses yang digunakan selama squid melayani request dari client dapat ditingkatkan, karena rekayasa rutin proses yang telah dibuat selalu melakukan koneksi ke basisdata dan kemudian memutusnya setiap kali mengambil data pada basisdata MySQL.
- Jenis konfigurasi lain tentang hak akses user, seperti alamat ip koneksi yang diijinkan atau dilarang untuk digunakan client, bisa ditambahkan dalam basisdata MySQL.
- Keamanan antarmuka berbasis web dapat lebih ditingkatkan. Salah satunya dapat menggunakan Socket Secure Layer (SSL) dalam setiap proses yang dilakukan melalui antarmuka berbasis web, selain mengandalkan keamanan melalui proses login.

9. DAFTAR PUSTAKA

1. Chadd, Adrian; Collins, Robbert; Nordstrom, Henrik; Rousskov, Alex; Wessels, Duane. : "Squid Web Proxy *Cache*". 1998; Available from: http://www.squid-*cache*.org. Accessed April 10, 2004.
2. Matthew, Neil; Stones, Richard. "Proffesional linux programming". Wrox Press Ltd.; 2000.
3. "MySQL reference manual", MySQL Documentation Team 2004.
4. "Pattern seen in the Squid Architecture". 2001; Available from: http://wiki.cs.uiuc.edu/cs427/Pattern+seen+in+the+Squid+Architecture. Accessed Maret 30, 2004.
5. Person, Oskar. "Squid user's guide version 1.1". Qualica Technologies (Pty) Ltd.; 2003.
6. "PHP manual version 2", PHP Documentation Group 2001.
7. Rivest, L., Ronald. "RFC 1321 The MD5 Message-Digest Algorithm". MIT Laboratory for Computer Science and RSA Data Security, Inc.; 1992.